Assignment – 1

Name – Prabhnoor Singh

Branch – ENC

Subgroup – NC3

Subject – Artificial Intelligence

Submitted to: Mrs. Tanvi Dovedi

Python Basic

1. Write a program to demonstrate different number datatypes and arithmetic operations (addition, subtraction, multiplication, division, and exponent) in python.

   Code –

   ```python
   n1 = int(input ("Enter First Number:"))
   n2 = int (input ("Enter Second Number:"))
   o = input ("Operation:")

   if o == "+" :
       result = n1 + n2
   elif o == "-" :
       result = n1 - n2
   elif o == "/" :
       result = n1 / n2
   elif o == "*" :
       result = n1 * n2
   elif o == "**" :
       result = n1 ** n2

   print (n1,o,n2, ":", result)
   ```

   Output:

   ```
   Enter First Number: 4
   Enter Second Number: 2
   Operation: /
   4 / 2 : 2.0
   ```

2. Write a program to implement all the usual operators for Boolean logic in python.

   Code –

   ```python
   n1 = int(input ("Enter First Number:"))
   n2 = int (input ("Enter Second Number:"))

   if n1 > 0 and n2 > 0 :
       print("Both the numbers are greater than 0")
   else:
       print("Atleast one number is not greater than 0")

   if n1 or n2 :
           print ("Atleast one of the numbers has boolean value as True")
   else :
               print ("Both the numbers have boolean value as False")

   if not n1:
       print("Boolean value of n1 is False")
   else:
       print ("Boolean value of n1 is True")
   ```

   Output:

   ```
   Enter First Number: 1
   Enter Second Number: 2
   Both the numbers are greater than 0
   Atleast one of the numbers has boolean value as True
   Boolean value of n1 is True
   ```

3. Write a program to perform certain operations on strings (print a string, concatenation of two strings, capitalize a string, convert a string to uppercase, right-justify, centre a string, replace all instances of one substring with another, accessing substring from a given string).

Code –

```python
var1 = str(input("Variable 1 :"))
var2 = str(input("Variable 2 :"))

print (var1)
print (var2)
print ("Concatenation of two strings :")
var3 = var1 + var2
print (var3)
print ("Capitalize a string :")
print (var3.capitalize())
print ("Convert a string to uppercase :")
print (var3.upper())
print ("Right-justify :")
print (var3.rjust(15))
print ("Center a string :")
print (var3.center(10))
print ("Replace all instances of one substring with another :")
print (var3[0:4])
print ("Accessing substring from a given string :")
print (var3.replace(var1,var2))
```

Output:

```
Variable 1 : Learning
Variable 2 : Python
Learning
Python
Concatenation of two strings :
LearningPython
Capitalize a string :
Learningpython
Convert a string to uppercase :
LEARNINGPYTHON
Right-justify :
 LearningPython
Center a string :
LearningPython
Replace all instances of one substring with another :
Lear
Accessing substring from a given string :
PythonPython
```

4. Write a program to create a list with different data types, access elements of lists, slicing, append and remove elements from list.

Code –

```python
list = [2, "Python", 3.14, "NUM12"]
print ("Acess elements of list :")
print(list)
print ("Slicing elements from list :")
print (list [1:3])
print("Appending element into list :")
list.append(-5)
print(list)
print("Removing element from list :")
list.remove(3.14)
print(list)
```

Output:

```
Acess elements of list
[2, 'Python', 3.14, 'NUM12']
Slicing elements from list
['Python', 3.14]
Appending element into list
[2, 'Python', 3.14, 'NUM12', -5]
Removing element from list
[2, 'Python', 'NUM12', -5]
```

5. Write a program to create a dictionary with different data types, access elements of dictionary, add and remove elements from dictionary.

Code –

```python
dictionary = {"Num1" : 2, "Name1" : "Apple", "Float1" : 3.14}
print ("Acess elements of dictonary :")
print(dictionary)
print("Adding element into list :")
dictionary["Num2"] = -5
dictionary["Name2"] = "Orange"
print(dictionary)
print("Removing element from list :")
del dictionary["Num1"]
del dictionary["Float1"]
print(dictionary)
```

Output:

```
Acess elements of dictonary
{'Num1': 2, 'Name1': 'Apple', 'Float1': 3.14}
Adding element into list
{'Num1': 2, 'Name1': 'Apple', 'Float1': 3.14, 'Num2': -5, 'Name2': 'Orange'}
Removing element from list
{'Name1': 'Apple', 'Num2': -5, 'Name2': 'Orange'}
```

6. Write a program to create a set with different data types, check for an element, add and remove elements from the set.

Code –

```python
set = {2, "Apple", 3.14, "Coding on Anaconda"}
print ("Acess elements of set :")
print(set)
set.add (-8)
set.add ("Kiwi")
print("Adding element into list :")
print(set)
set.remove (2)
set.remove (-8)
print("Removing element into list :")
print (set)
```

Output:

```
Acess elements of set :
{'Apple', 2, 3.14, 'Coding on Anaconda'}
Adding element into list :
{2, 3.14, 'Apple', 'Kiwi', 'Coding on Anaconda', -8}
Removing element into list :
{3.14, 'Apple', 'Kiwi', 'Coding on Anaconda'}
```

7. Write a program to create a tuple with different data types, create a dictionary with tuple keys.

Code –

```
Tuple = {(3.2, 6.4, 9.6) : 'Float',("Red", "Yellow", "White") : 'Colours', (2, 4, 6) : 'Integer'}
Tuple
```

Output:

```
{(3.2, 6.4, 9.6): 'Float',
 ('Red', 'Yellow', 'White'): 'Colours',
 (2, 4, 6): 'Integer'}
```

8. Write a program to print the sign of number.

Code –

```
n = int(input('Enter the value :'))
if n > 0 :
    print("The number is positive and it's sign is '+'")
elif n < 0 :
        print("The number is negative and it's sign is '-'")
else :
    print("The number is Zero")
```

Output:

```
Enter the value : -5
The number is negative and it's sign is '-'
```

9. Using a conditional, write the relu function defined as follows:

$$relu = \begin{cases} x, if\ x \geq 0 \\ 0, otherwise \end{cases}$$

Code –

```
def relu (num):

    if num >= 0 :
        return num

    else :
        return 0

x = int(input('Enter the value :'))
print("The relu is", relu(x))
```

Output:

```
Enter the value : 18
The relu is 18
```

10. (a) Write a python program to print prime numbers less than 20.

Code –

```
start = 1
end = 20
for n in range (start,end+1) :
    count = 0
    for i in range (2,int((n*0.5)+1)):
        if (n%i) == 0 :
            count = count +1
            break
    if (count == 0 and n>1):
        print ("%d  " %n, end = '')
```

Output:

```
2  3  5  7  11  13  17  19
```

(b) Write a python program to find factorial of a number using recursion.

Code –

```
def fact(num) :
    if num ==1 :
        return 1
    else :
        return (num*fact(num-1))

n = int(input("Enter number whose factorial you want to find: "))
print ("Factorial: ", fact(n))
```

Output:

```
Enter number whose factorial you want to find:  9
Factorial:  362880
```

(c) Write a program that inputs a text file. The program should print all the unique words in the file in alphabetical order.

Code –

```
file = open('sample.txt', 'w')
file.write('My name is Prabhnoor Singh')
file.close()
filename = input ("Enter the filename: ")
unique_words = set()

try :
    with open(filename, 'r') as file:
        for line in file :
            words = line.strip().split()
            unique_words.update(words)

            sorted_words = sorted(unique_words)
            print ("Unique words in alphabetical order: ")
            for word in sorted_words:
                print(word)

except FileNotFoundError:
    print("File not found.")
```

Output:

```
Enter the filename: sample.txt
Unique words in alphabetical order:
My
Prabhnoor
Singh
is
name
> |
```

11. Write a python program to square each element of a list using list comprehension, and of a dictionary using dictionary comprehension.

Code –

```python
list = [1,2,3,4,5,6]
list_sqr = [i*i for i in list]
print ("Original list: ", list)
print ("Squared elements of a list using list comprehension: ", list_sqr)

dictionary_sqr = {j:j*j for j in list}
print ("Squared elements of a dictionary using dictionary comprehension: ", dictionary_sqr)
```

Output:

```
Original list:  [1, 2, 3, 4, 5, 6]
Squared elements of a list using list comprehension:  [1, 4, 9, 16, 25, 36]
Squared elements of a dictionary using dictionary comprehension:  {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36}
```